UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/814,774	03/30/2004	David K. Parker	02453.0033.NPUS00	8909

27194 7590 12/03/2009

HOWREY LLP-CA
C/O IP DOCKETING DEPARTMENT
2941 FAIRVIEW PARK DRIVE, SUITE 200
FALLS CHURCH, VA 22042-2924

EXAMINER	
AHMED, SALMAN	

ART UNIT	PAPER NUMBER
2476	

MAIL DATE	DELIVERY MODE
12/03/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/814,774	PARKER ET AL.	
	Examiner	Art Unit	
	SALMAN AHMED	2476	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10/4/2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-3,6-15,18-22,24 and 28-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3,6-12, 14, 15,18-22,24 and 28-30* is/are rejected.

7)☒ Claim(s) *13* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

3. Claims 1-3, 6-8, 11, 12, 14, 15, 18-20, 24, 28 and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jennings et al. (US PAT 5710923, hereinafter Jennings) in view of York et al. (US PAT 6002881, hereinafter York) and Wu (US PAT PUB 2003/0204840).

In regards to claim 1, Jennings teaches a *processor readable, physical medium, encoding a data structure* (figure 2 element 16 or figure 5 element 16') *for supporting one or more packet modification operations, the data structure comprising: a first pointer* (figure 2, 16b or figure 5, 16b), *entered in the data*

structure, to a sequence of commands (figures 2 or 5 element 18a), *executable by a processor, implementing one or more packet modification operations and stored,* (figures 2 or 5, system memory 18), *with more than one command stored in a single addressable entry of the stored sequence* (column 1 lines 45-50, Procedures are used in programming languages to describe sequences of operations (satisfying limitation more than one command) on data. In object oriented languages (e.g., C++), a data object is implicitly referenced by a procedure. The procedure represents a certain action to be performed on the data object, and parameters may be supplied to the procedure (by the caller) to qualify the exact details of the operation), *in a first memory area* (figures 2 or 5 element 18a); *and a second pointer* (figures 2 or 5, 16c), *also entered in the data structure, to a of data or mask items*, *stored, with more than one data or mask item* (figures 2 or 5, more than one data or mask item is satisfied by Application specific data structure and frame memory) *stored in a single addressable entry of the stored data or mask* (figures 2 or 5, 16c*), in a second memory area distinct from the first* (figures 2 or 5, 18 a is different from 18b), *for use by the processor in executing the one or more commands*; *wherein the commands in the sequence specify performing the one or more packet modification operations using, as operands or as masks for operands, the data or mask items in the data or mask* (column 1, lines 34-44, column 2, lines 55-67, column 5 lines 13-22, FIG. 2 depicts a typical active messaging model in which an active message 16 comprises a message header 16a, instruction pointer (.mu.IP) 16b, frame pointer (.mu.FP) 16c, and one or more words of additional data or parameters in a

segment 16d of the message. As shown, the instruction pointer 16b identifies an application specific procedure 18a in system memory 18, and the frame pointer 16c identifies an application specific data structure 18b in system memory. The application specific procedure 18a employs various parameters in the segment 16d of the message 16. depicted in FIG. 2, each thread descriptor includes an instruction pointer and a frame pointer. The instruction pointer points to a particular instruction held in the code section 26 of the local memory of the processing element, and the frame pointer points to the beginning of a particular frame of memory locations in the frames section 24 of the local memory. When a thread descriptor enters the pipeline 36, its instruction pointer is used to locate a particular instruction in the code section 26 of local memory. The processing pipeline 36 then obtains the appropriate operands for that instruction by referring to the frame pointer, which is used to locate a particular frame held in the frames portion 24 of local memory. The frame pointer points to the beginning of the specified frame, and the addresses held in the operand field of the fetched instruction are used as offsets to locate the memory locations where the desired operands are held. The processing pipeline 36 then performs the desired function specified by the instruction, and then stores the result in the frames portion of local memory. After the result operand is stored, a new thread descriptor for the same thread is placed in the queue 22. FIG. 5 depicts a model of an active messaging system in accordance with the present invention. As shown, the inventive active messaging system employs an active message 16' comprising an instruction pointer (.mu.IP) 16b, frame pointer (.mu.FP) 16c, and

Local Parameters pointer 16e. In addition, the system includes one or more words of additional data or parameters 16d' stored in memory mapped device registers or system memory. The instruction pointer 16b identifies an application specific procedure 18a in system memory 18, and the frame pointer 16c identifies an application specific data structure 18b in system memory). In regards to processor, Jennings teaches column 8, lines 64-67, with the present invention, a .mu.thread retains exclusive control of a processor (and no other .mu.thread executes on that processor) until the .mu.thread explicitly terminates itself.

Jennings does not explicitly teach data being in burst format.

York in the same or similar field of endeavor teaches data being in a burst format (column 1 lines 35-42).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate in Jennings' system/method the steps of using burst format for data as suggested by Jennings. The motivation is that (as suggested by York, column 1 lines 35-42) the advantage to use burst mode is that such method transfers in which a start address is provided to the memory which then returns data words from a sequence of adjacent memory locations; thus with burst mode transfers efficiency gains are achieved. Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces/market place incentives if the variations are predictable to one of ordinary skill in the art.

Jennings and York do not explicitly teach instructions/commands being in packed format.

Wu in the same or similar field of endeavor teaches instructions/commands/data being in packed format (paragraph 0052).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate in Jennings and York's system/method the steps of instructions/commands/data being in packed format as suggested by Wu. The motivation is that (as suggested by Wu, paragraph 0052) by including a packed instruction set in a standard microprocessor instruction set, packed data instructions can be easily incorporated into existing software (previously written for the standard microprocessor instruction set). Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces/market place incentives if the variations are predictable to one of ordinary skill in the art.

In regards to claims 11, 12 and 14 Jennings teaches retrieving from a memory a data structure (figure 2 element 16 or figure 5 element 16', mu.thread) corresponding to the data structure link (abstract, generating a .mu.thread comprising an instruction pointer, frame pointer, and Local Parameters pointer from a first node to a second node), the data structure comprising a first pointer (figure 2, 16b or figure 5, 16b), entered in the data structure, to a sequence of commands (figures 2 or 5 element 18a), for execution by a processor, implementing one or more packet modification operations and stored (figures 2

or 5, system memory 18), with more than one command stored in a single addressable entry in the stored sequence (column 1 lines 45-50, Procedures are used in programming languages to describe sequences of operations (satisfying limitation more than one command) on data. In object oriented languages (e.g., C++), a data object is implicitly referenced by a procedure. The procedure represents a certain action to be performed on the data object, and parameters may be supplied to the procedure (by the caller) to qualify the exact details of the operation), in a first memory area (figures 2 or 5 element 18a), and a second pointer (figures 2 or 5, 16c), also entered in the data structure, to data or mask items, stored, with more than one data or mask item (figures 2 or 5, more than one data or mask item is satisfied by Application specific data structure and frame memory) stored in a single addressable entry in the stored data or mask (figures 2 or 5, 16c), in a second memory area distinct from the first (figures 2 or 5, 18 a is different from 18b), for use by the processor in executing the one or more commands; retrieving from the first memory area the commands in the sequence by using the first pointer; retrieving from the second memory area the data or mask items by using the second pointer; and executing the commands in the sequence by the processor, thereby performing one or more packet modification operations on the packet using as operands or as masks for operands in the one or more packet modification operations, the data or mask items (column 1, lines 34-44, column 2, lines 55-67, column 5 lines 13-22, FIG. 2 depicts a typical active messaging model in which an active message 16 comprises a message header 16a, instruction pointer (.mu.IP) 16b, frame pointer

(.mu.FP) 16c, and one or more words of additional data or parameters in a segment 16d of the message. As shown, the instruction pointer 16b identifies an application specific procedure 18a in system memory 18, and the frame pointer 16c identifies an application specific data structure 18b in system memory. The application specific procedure 18a employs various parameters in the segment 16d of the message 16. depicted in FIG. 2, each thread descriptor includes an instruction pointer and a frame pointer. The instruction pointer points to a particular instruction held in the code section 26 of the local memory of the processing element, and the frame pointer points to the beginning of a particular frame of memory locations in the frames section 24 of the local memory. When a thread descriptor enters the pipeline 36, its instruction pointer is used to locate a particular instruction in the code section 26 of local memory. The processing pipeline 36 then obtains the appropriate operands for that instruction by referring to the frame pointer, which is used to locate a particular frame held in the frames portion 24 of local memory. The frame pointer points to the beginning of the specified frame, and the addresses held in the operand field of the fetched instruction are used as offsets to locate the memory locations where the desired operands are held. The processing pipeline 36 then performs the desired function specified by the instruction, and then stores the result in the frames portion of local memory. After the result operand is stored, a new thread descriptor for the same thread is placed in the queue 22. FIG. 5 depicts a model of an active messaging system in accordance with the present invention. As shown, the inventive active messaging system employs an active message 16'

comprising an instruction pointer (.mu.IP) 16b, frame pointer (.mu.FP) 16c, and Local Parameters pointer 16e. In addition, the system includes one or more words of additional data or parameters 16d' stored in memory mapped device registers or system memory. The instruction pointer 16b identifies an application specific procedure 18a in system memory 18, and the frame pointer 16c identifies an application specific data structure 18b in system memory). In regards to processor, Jennings teaches column 8, lines 64-67, with the present invention, a .mu.thread retains exclusive control of a processor (and no other .mu.thread executes on that processor) until the .mu.thread explicitly terminates itself.

Jennings does not explicitly teach data being in burst format.

York in the same or similar field of endeavor teaches data being in a burst format (column 1 lines 35-42).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate in Jennings' system/method the steps of using burst format for data as suggested by Jennings. The motivation is that (as suggested by York, column 1 lines 35-42) the advantage to use burst mode is that such method transfers in which a start address is provided to the memory which then returns data words from a sequence of adjacent memory locations; thus with burst mode transfers efficiency gains are achieved. Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces/market place incentives if the variations are predictable to one of ordinary skill in the art.

Jennings and York do not explicitly teach instructions/commands being in packed format.

Wu in the same or similar field of endeavor teaches instructions/commands/data being in packed format (paragraph 0052).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate in Jennings and York's system/method the steps of instructions/commands/data being in packed format as suggested by Wu. The motivation is that (as suggested by Wu, paragraph 0052) by including a packed instruction set in a standard microprocessor instruction set, packed data instructions can be easily incorporated into existing software (previously written for the standard microprocessor instruction set). Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces/market place incentives if the variations are predictable to one of ordinary skill in the art.

In regards to claims 3, 15 and 29, Jennings, York and Wu do not explicitly teach first memory and second memory are located in the same memory. It would have been obvious i.e. "Obvious to try" – choosing from a finite number of identified, predictable solutions, to one having ordinary skill in the art at the time the invention was made to modify Jennings, York and Wu's system/method with the steps of first memory and second memory being located in the same memory depending on system resources, system specification, network requirement, design choice etc. to implement an efficient system.

In regards to claims 6 and 18 Jennings teaches the data or mask items comprise data items and associated mask items, with a data item (figures 2 or 5, frame memory) stored adjacent (i.e. adjacent within 18b) to its associated mask item (figures 2 or 5, Application specific data structure).

In regards to claims 6 and 18 Jennings does not explicitly teach data being in burst format.

York in the same or similar field of endeavor teaches data being in a burst format (column 1 lines 35-42).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate in Jennings' system/method the steps of using burst format for data as suggested by Jennings. The motivation is that (as suggested by York, column 1 lines 35-42) the advantage to use burst mode is that such method transfers in which a start address is provided to the memory which then returns data words from a sequence of adjacent memory locations; thus with burst mode transfers efficiency gains are achieved. Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces/market place incentives if the variations are predictable to one of ordinary skill in the art.

In regards to claims 7 and 19, Jennings, York and Wu do not explicitly teach first memory and second memory are located in a memory implemented off chip in relation to te processor. It would have been obvious i.e. "Obvious to try" – choosing from a finite number of identified, predictable solutions, to one having ordinary skill in the art at the time the invention was made to modify Jennings,

York and Wu's system/method with the steps of first memory and second memory are located in a memory implemented off chip in relation to te processor depending on system resources, system specification, network requirement, design choice etc. to implement an efficient system.

In regards to claims 8 and 20, Jennings, York and Wu do not explicitly teach first memory and second memory are located in a memory implemented on chip in relation to te processor. It would have been obvious i.e. "Obvious to try" – choosing from a finite number of identified, predictable solutions, to one having ordinary skill in the art at the time the invention was made to modify Jennings, York and Wu's system/method with the steps of first memory and second memory are located in a memory implemented on chip in relation to te processor depending on system resources, system specification, network requirement, design choice etc. to implement an efficient system.

Claims 24 and 28 teaches all the limitations of claim 1 or 11 above and are thus rejected using the same rationale. In regards to processor, Jennings teaches column 8, lines 64-67, with the present invention, a .mu.thread retains exclusive control of a processor (and no other .mu.thread executes on that processor) until the .mu.thread explicitly terminates itself.

Claim 29 teaches all the limitations of claims 3 or 15 above and are thus rejected using the same rationale.

4. Claims 9 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jennings et al. (US PAT 5710923, hereinafter Jennings), York

et al. (US PAT 6002881, hereinafter York) and Wu (US PAT PUB 2003/0204840) in view of Teich et al. (US PAT 5943493, hereinafter Teich).

In regards to claims 9 and 21 Jennings, York and Wu teaches all the limitations of claims 1 and 11 above, but do not explicitly teach data structure comprises plurality of pointers, each to a sequence of commands implementing one or more packet modification operations.

Teich in the same or similar field of endeavor teaches data structure comprises plurality of pointers, each to a sequence of commands implementing one or more packet modification operations (abstract).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate in Jennings and York and Wu's system/method the steps of data structure comprises plurality of pointers, each to a sequence of commands implementing one or more packet modification operations as suggested by Teich. The motivation is that having multiple pointers within a data structure or table, pointing to multiple instructions, enable a system to seamlessly and efficiently execute multiple instructions with less usage of processing resources. Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces/market place incentives if the variations are predictable to one of ordinary skill in the art.

5. Claims 10 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jennings et al. (US PAT 5710923, hereinafter Jennings), York

et al. (US PAT 6002881, hereinafter York) and Wu (US PAT PUB 2003/0204840) in view of Sharangpani et al. (US PAT 5367650, hereinafter Sharangpani).

In regards to claims 10 and 22, Jennings, York and Wu teaches all the limitations of claims 1 and 11 above, but do not explicitly teach data structure comprises plurality of pointers, each to a data or mask item.

Sharangpani in the same or similar field of endeavor teaches data structure comprises plurality of pointers, each to a data or mask item (abstract).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate in Jennings and York and Wu's system/method the steps of data structure comprises plurality of pointers, each to a data or mask item as suggested by Sharangpani. The motivation is that having multiple pointers within a data structure or table, pointing to multiple data, enable a system to seamlessly and efficiently manipulate multiple data with less usage of processing resources. Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces/market place incentives if the variations are predictable to one of ordinary skill in the art.

1. Claim 30 is rejected under 35 U.S.C. 103(a) as being unpatentable over Jennings et al. (US PAT 5710923, hereinafter Jennings), York et al. (US PAT 6002881, hereinafter York) and Wu (US PAT PUB 2003/0204840) in view of Corby, Jr. et al. (US PAT 5524258, hereinafter Corby).

In regards to claim 30, Jennings, York and Wu teach the one or more commands and all the limitations of claim 24, but do not explicitly teach a pipeline processor core configured to retrieve commands in a sequence in a first stage, and execute the commands in a sequence in one or more subsequent stages.

Corby in the same or similar field of endeavor teaches a real-time data processing system employs a control computer which defines a pre-processing arrangement of data channels to speed processing, and an arrangement of output data channels to provide a desired output format. The data channels are samples and arranged into a data packet which is passed to an array of digital signal processors (DSPs) arranged in a series of stages, with at least one DSP per stage. A front-end DSP receives the data packet and appends a control field having commands addressed to specific DSPs to the data packet along with adding a monitor field. The DSPs monitor the control field for commands addressed to it and then executes those. The status of the operation is written in the monitor field and the data packet is passed to DSPs of the next stage for `pipelined` processing. DSPs of the last stage collect the process portions of the data packet, assemble them according to the desired output format and pass on the completed data packet. The system control computer may monitor the monitor field of any data packet and determine the health of each DSP (abstract).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to modify Jennings, York and Wu's system/method with the steps of a a pipeline processor core configured to retrieve commands in a sequence in a first stage, and execute the commands in a sequence in one or

more subsequent stages as suggested by Corby. The motivation is that (as suggested by Corby lines 45-55, paragraph 1) a particularly useful way of arranging the many processors is to organize them as cascaded groups of processors, with each group forming a stage operating within the same time period. The outputs of a given stage form the inputs to the succeeding stage. The total processing task is decomposed into a finite set of sequential operations. This is called pipelining. These tasks are distributed over the stages of processor array; thus implementing an efficient processing system. Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces/market place incentives if the variations are predictable to one of ordinary skill in the art.

Allowable Subject Matter

6. Claim 13 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Response to Arguments

7. Applicant's arguments see pages 6-14 of the Remarks sections, filed 10/4/2009, with respect to the rejections of the claims have been fully considered and are moot in view of new ground of rejections presented in this office action.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SALMAN AHMED whose telephone number is (571)272-8307. The examiner can normally be reached on 9:00 am - 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh can be reached on (571)272-3795. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Salman Ahmed/
Primary Examiner, Art Unit 2476